Specification

In this lab we will explore 2-dim arrays and see how they can be used to represent a drawing as a bit map. Each pixel of the drawing will be an element of the array with 0 meaning one color and 1 meaning a different color. These may be black and white or any other 2 colors. We will create several modules:

1. **main** to setup the GUI and to call functions to accomplish these goals:
2. Load bitmap image into a *global* array.
3. Draw the image on the Cairo window from data in array.
4. Modify the data in the array to change the orientation of image:
    - horizontal flip
    - vertical flip
    - mirror image across diagonal
    - invert the 2 colors
    - let user choose different colors
    - let use choose a different image
5. Respond to button press to let user choose between modifications to image.

Here is an example bitmapped image: arrow.pbm

Use this for one of your images, and find or create a second one as well.

Read the rubric and the descriptions of each requirement.

Remember that the **PDF** file is what is graded so you must verify that
EVERY line of source code is displayed in the pdf and that each one has comments written by YOU to explain the code. Those comments are worth a *high percentage* of the grade.

Upload and submit the tgz file with all files from your working directory as well as the pdf!

Analysis

Input a bit map(one of the two) as a string from our pbm file.(User's choice of colors, way of modify...)

Preocess Store the string into the global array, where 0 means one color and 1 means a different color. When the user click the buttons, modify the image by modify (the data which stored in) the global array based on which button the user clicked.(respond to user's input)

Output Show a bit map which is modified by our functions and user's choices ,and the program uses our 2-dim array to draw it.

Design

- We have a function for flip vertically named flipv, it gets array of integer as input and then the change the position of the bit from (x,y) into (x,-y)in the coordination.
- We have a function for flip color of the image named flipc, it gets array of integer as input and then change all the value in it from 1 into 0, all the 0 into 1.
- We have a function for flip horizontally named fliph, it gets array of integer as input and then use the change the position of the bit from (x,y) into (-x,y)in the coordination.
- We have two function for flip diagonally named flipd1 and flipd2, they get array of integer as input and then swap all the value in it symmetrically along either diagonal.
- We have a function named drawCB ,which uses cairo for choose color, add text, make the window....It will help us create the window.

- We have a function named goCB, it use fl_choice() to create buttons and let user do some choices to modify the image :
- horizontal flip
- vertical flip
- mirror image across diagonal
- invert the 2 colors
- let user choose different colors
- resize it by 10 or 15
- We have a function for load the image named load_image, it gets the pbm file as a string and then store the value in the string into the array(which can be shown as a bit map).
- We have a main function which calls the fl_choice to let user choose and load a image, make the window(title,size), and calls functions to create buttons and the functions we created,etc.

Implementation

```
#include "config.h"
#include <FL/Fl_Cairo_Window.H>
#include <FL/Fl_Button.H>
#include <FL/fl_ask.H>
#include <iostream>
#include <fstream>
#include <FL/Fl_Color_Chooser.H>
const int WIDTH = 400;
const int HEIGHT = 400;
int scale = 2;
const int COLUMNS = 32;  //depend on actual size of image
const int ROWS = 32;
int arrow[ROWS][COLUMNS]; //integer

//fuchsia rgb(100%,0%,100%)
double r=1;double g=0; double b=1;

//
```

```
//flip the image vertically
void flipv(int a[ROWS][COLUMNS])
{
        for(int row = 0; row < ROWS/2; row++)
        {
                for(int column = 0; column < COLUMNS; column++)
                {
                        std::swap(a[row][column],
                        a[ROWS - 1 - row][column]);
                        //change the pixel in the coordination from:
                        //(x,y)->(x,-y)

                }
        }
}

//
```

```
//flip the color of the image
void flipc(int a[ROWS][COLUMNS])
{
        for(int row = 0; row < ROWS; row++)
        {
                for(int column = 0; column < COLUMNS; column++)
                {
                        a[row][column] = abs(a[row][column] - 1);
                        //chang the data stored in the array from:
                        //0->1
                        //1->0
                }
        }
}

//
```

```
//flip the image horizontally
void fliph(int a[ROWS][COLUMNS])
{
        for(int column = 0; column < COLUMNS/2; column++)
        {
                for(int row = 0; row < ROWS; row++)
                {
                        std::swap(a[row][column],
                        a[row][COLUMNS - 1 - column]);
                        //change the pixel in the coordination from:
                        //(x,y)->(-x,y)
                        //condition ensures the array won't
                        //be swap twice

                }
        }
}

//
```

```
//flip the image diagonally(in this way-> / )
void flipd1(int a[ROWS][COLUMNS])
{
        for(int row = 0; row < ROWS; row++)
        {
                for(int column = 0; column < (COLUMNS-row); column++)
                {
                        std::swap(a[row][column],
                        a[COLUMNS-1-column][ROWS-1-row]);
                        //change the pixel in the coordination from:
                        //(x,y)->(-x,-y)
                        //condition ensures loop stop swapping when
                        //reach the diagonal
                }
        }
}

//
```

```
//flip the image diagonally(in this way-> \ )
void flipd2(int a[ROWS][COLUMNS])
{
        for(int row = 0; row < ROWS; row++)
        {
                for(int column = 0; column < row; column++)
                {
                        std::swap(a[row][column],a[column][row]);
                        //(x,y)->(y,x)
                        //condition ensures loop stop swapping when
                        //reach the diagonal
                }
        }
}

//
```

```
//Choose color, make the window, add text...
void drawCB(Fl_Cairo_Window* cw,cairo_t* cr)
{
        cairo_set_source_rgb(cr,1,1,1); //white
        cairo_paint(cr);

        cairo_set_source_rgb(cr,1,0,0); //red
        cairo_rectangle(cr,HEIGHT-scale,
        WIDTH-scale,scale,scale);  //x,y,w,h
        cairo_stroke(cr); // to outline the button


        cairo_set_source_rgb (cr, 1, 0, 0);//  red  rgb
        cairo_select_font_face (cr, "Georgia",
                CAIRO_FONT_SLANT_NORMAL, CAIRO_FONT_WEIGHT_BOLD);
        cairo_set_font_size (cr, 12);
        cairo_move_to (cr, cw->x() , cw->h()/6 );
        cairo_show_text (cr, "Click Anywhere");
        // put label on window

        cairo_translate(cr,cw->w()/2-COLUMNS,cw->h()/2-ROWS);
        //decide the position of the text

        for(int row = 0; row < ROWS; row++)
                for(int column = 0; column < COLUMNS; column++)
                {
                        if(arrow[row][column] == 1)
                        {
                                cairo_set_source_rgb(cr,r,g,b);
                                //fuchsia rgb(100%,0%,100%)
                        }
                        else
                        {
                                cairo_set_source_rgb(cr,0,0,0);
                                //black
                        }
                                                                //x,y,w,h
                        cairo_rectangle(cr,column*scale,
                        row*scale,1*scale,1*scale);
                        cairo_fill(cr);
                }
        }

        //
```

```
//create buttons to let user choose options
//in order to modify the image
void goCB(void*,void*)
{
        switch ( fl_choice("Choose option:",
        "menu","by 2",
        "resize by 3"))
        {
        case 0:
                  switch(fl_choice("Choose option:",
                  "color","flip",nullptr))
                  //both nullptr and 0 work
                  //no enough space in pdf
                              {
                              case 0:
                                      switch(fl_choice("color?","inverse",
                                      "choose",nullptr))
                                      //both nullptr and 0 work
                                      {
                                              case 0:
                                                      flipc(arrow);
                                                      //flip the color
                                                      break;
                                              case 1:
                                                      fl_color_chooser(
                                                            "choose a color",
                                                            r,g,b);
                                                            //choose color
                                                            break;
                                                  }
                                        break;
                              case 1:
                                        switch(fl_choice("How?",
                                        "horizontal","vertical",
                                        "diagonal"))
                                        //no enough space in pdf
                                                  {
                                                            case 0:
                                                                      fliph(arrow);
                                                  //flip horizontally
                                                                      break;
                                                            case 1:
                                                                      flipv(arrow);
                                                  //flip vertically
                                                                      break;
                                                            case 2:
                                                                      switch
                                                                      (fl_choice
                                                                      ("Which?",
                                                                      "/","\\",
                                    nullptr))
/*
no space in the pdf...
I have no choice...
In order to let them visible,
I deleted some spaces in my code
*/
{
        case 0:
                        flipd1(arrow);
                                /*flip the
                                image
                                diagonally(/)
                                */
                        break;
        case 1:
                        flipd2(arrow);
                                /*flip the
                                image
                                diagonally(\)
                                */
                        break;
                                        }
                                break;
                        }
```

```
                                        break;
                                }
                                break;
                case 1: scale = 4; break;//resize by 2
                case 2: scale = 6; break;//resize by 3
        }
}

//
```

```
//translate the image(stored in a string) into
//an array(which can be showed as a bit map)
void load_image(std::string f)
{
        std::ifstream ifs(f);//get the string
        if(not ifs) std::cerr << "File not found" <<std::endl;
        // read and ignore the P1 and width and height
        //                    P    1    32  32
        char c; int i; ifs >> c >> i >> i >> i;
        std::cout << c << " " << i << std::endl;
        for(int row = 0; row < ROWS; row++)
                for(int column = 0; column < COLUMNS; column++)
                {
                        ifs >> arrow[row][column];
                        //now we get the array
                }
}

//
```

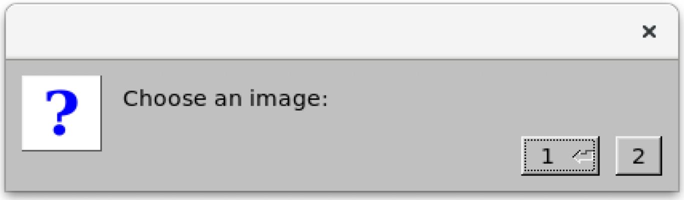
```
//create buttons to let user choose a image,calls
//function to load it, create the whole window,
//calls functions we have created to achieve our goal
int main()
{
        /*
        another way to access the goal:
        if(fl_ask("Do you want the first image?"))
        {load_image("102arrow.pbm");}
        else
        {load_image("arrow2.pbm");}
        */
        switch ( fl_choice("Choose an image:","2","1",nullptr))
        {
                case 0:
                                load_image("102arrow.pbm");
                                //load the first one
                                break;
                case 1:
                                load_image("arrow2.pbm");
                                //load the second one
                                break;
        }
        Fl_Cairo_Window cw(WIDTH,HEIGHT);
        cw.color(FL_WHITE);
```

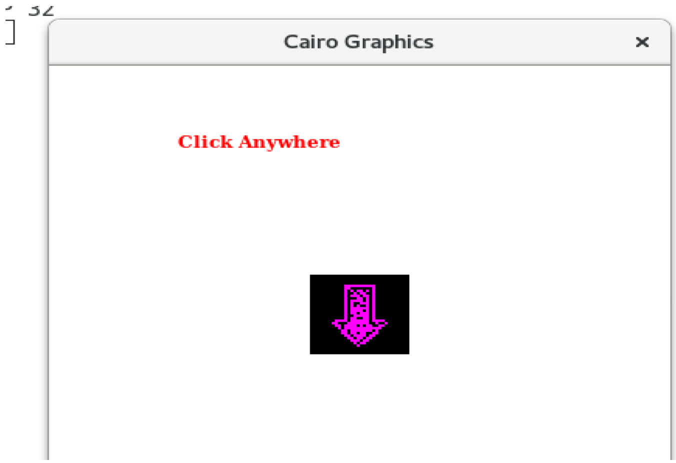
```
          cw.label("Cairo Graphics");
          cw.set_draw_cb(drawCB);
          Fl_Button b(cw.x(),cw.y(),cw.w(),cw.h());
          b.callback((Fl_Callback*) goCB);
          cw.show();
          return Fl::run();
}
```

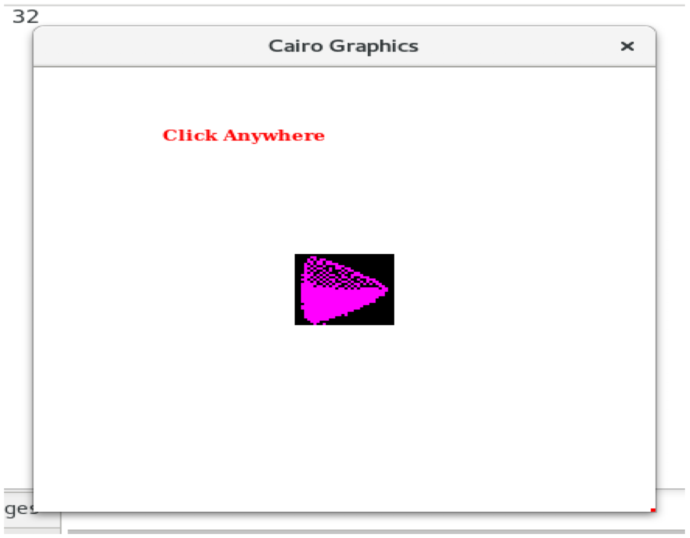
Test

Testcase of choosing a differntent image(it runs well)

- When the user execute our program,the user will see our first window like that,which tells the user to select an image.After the user selected, the one the user selected will be loaded into our program and shows in the window.
- (In all the testcases below, I only choose one of the two images as a sample)

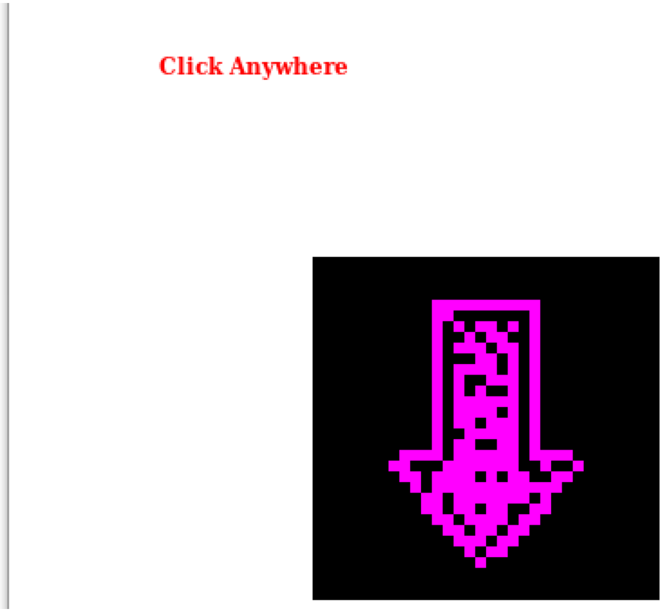


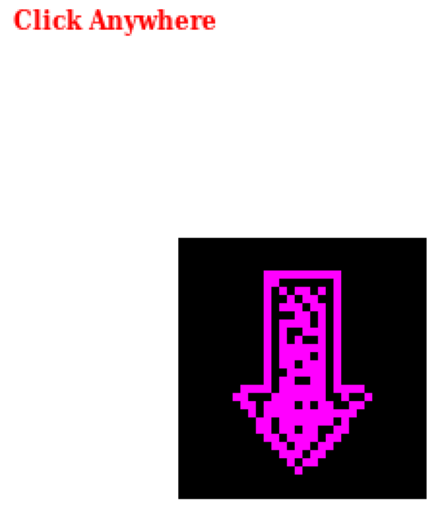


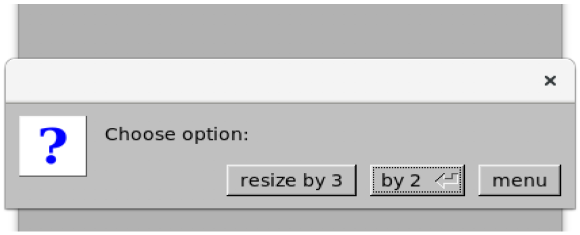

32
Cairo Graphics
Click Anywhere
ges

Testcase of resize by 2 or 3(it runs well)

- When the user execute our program,select an image,the user can click either of these two resize buttons to resize the image by 2 or 3.

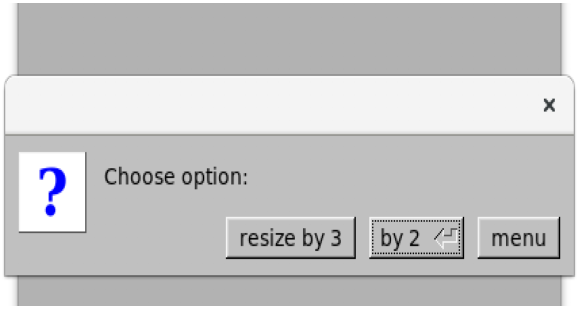


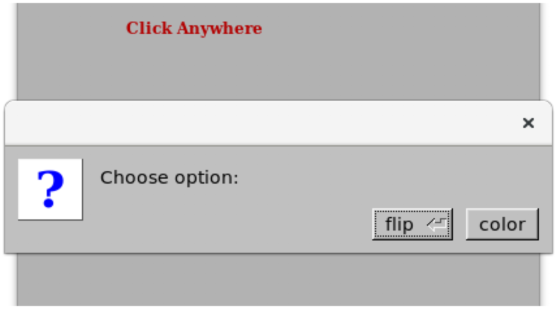



Choose option:

Testcase of enter the menu(it runs well)

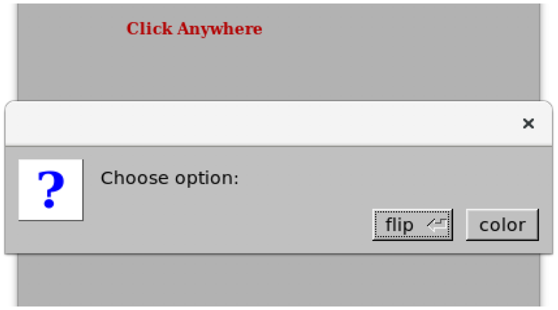
- When the user execute our program,select an image,the user can click the menu button to enter the menu,where the user can modify the image.

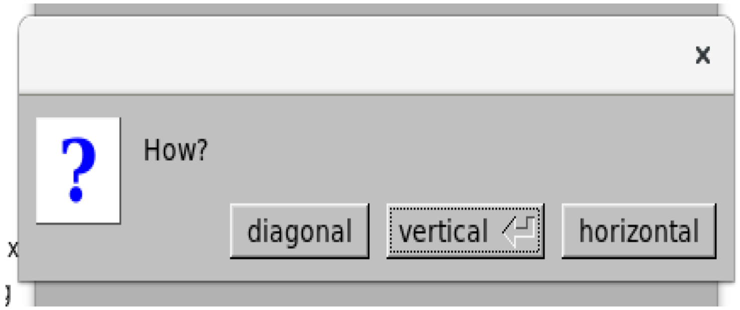




Testcase of flip vertically(it runs well)

- When the user execute our program,select an image,enter the menu,the user can click one of the three buttons to modify the image.If the user choose flip and then choose vertical,the image will be flipped vertically.

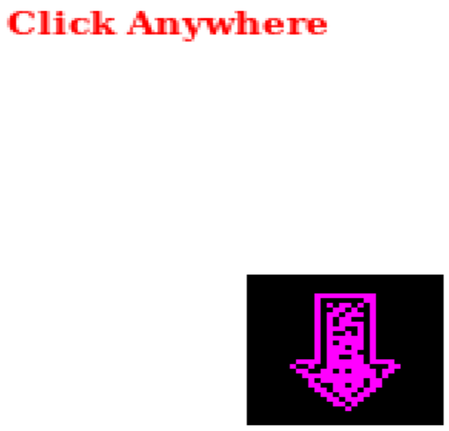


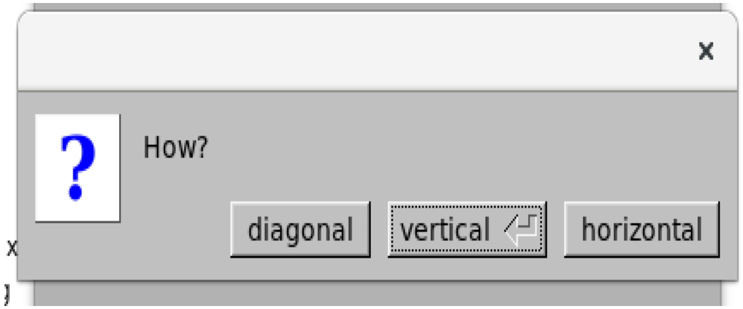


Click Anywhere

Testcase of flip horizontally (it runs well)

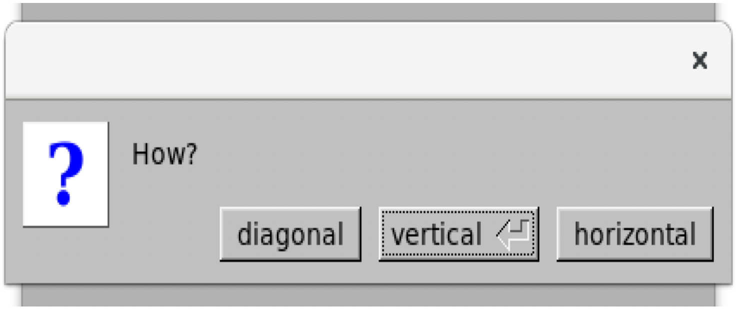
- When the user execute our program,select an image,enter the menu,the user can click one of the three buttons to modify the image(flip or change color).If the user choose flip and then choose horizontal,the image will be flipped horizontally.

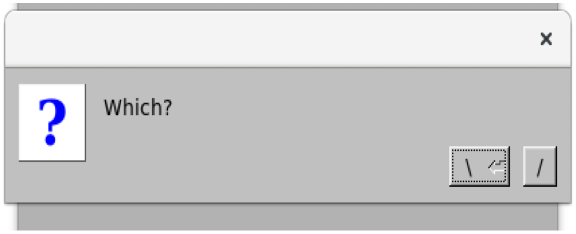



How?

Testcase of mirror image across diagonal(it runs well)

- When the user execute our program,select an image,enter the menu,the user can click one of the three buttons to modify the image(flip or change color).If the user choose flip and then choose diagonal, and one of the two diagonals, the image will be flipped horizontally by that diagonal.



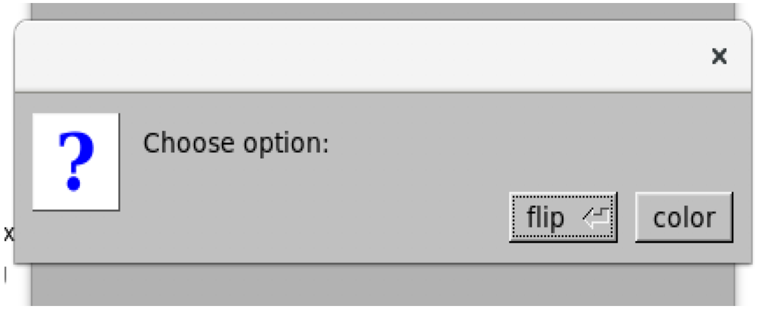


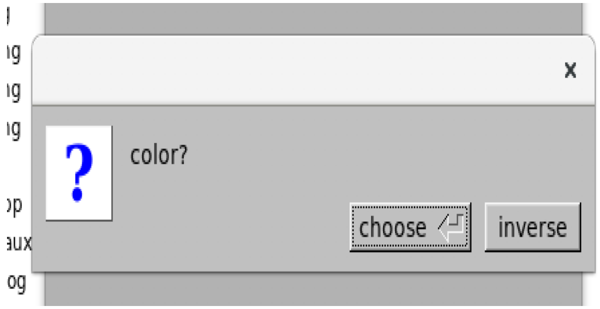
Click Anywhere

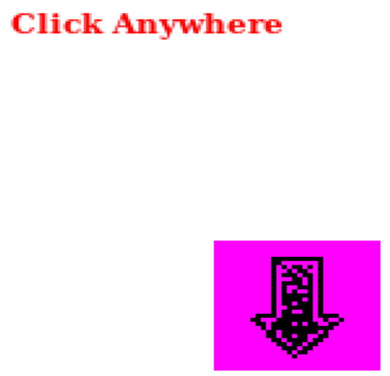
Click Anywhere

Testcase of invert the 2 colors(it runs well)

- When the user execute our program,select an image,enter the menu,the user can click one of the three buttons to modify the image(flip or change color).If the user choose color and then choose inverse,the 2 colors will be inverted.

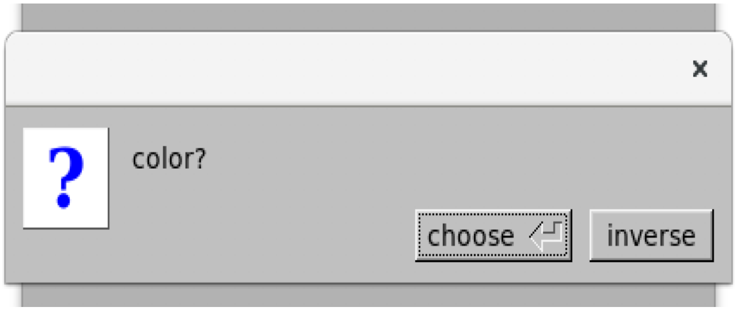




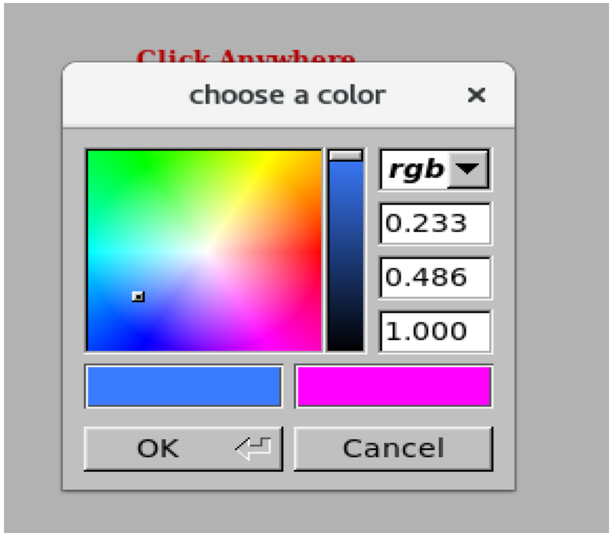


Testcase of choose colors(it runs well)

- When the user execute our program,select an image,enter the menu,the user can click one of the three buttons to modify the image(flip or change color).If the user choose color and then choose choose,the user can choose the color of the image.



color?
choose
inverse



Click Anywhere
choose a color
rgb
0.233
0.486
1.000
OK
Cancel

Click Anywhere

Conclusion of my testcases

As we can see in my testcases, all of my codes run well.

I successfully acheived the goal of this lab assignment!